as they are stored linearly in the array Xbar. This storage mode extends the usefulness of the routines by including those versions of FORTRAN which limit the number of subscripts to three or four. With linear representation, the analysis of N-way designs is limited only by the storage capacity of the user's computer center.

The computed cell means are passed as an argument to the sequence of subroutines listed above. The user need only assign values to the arguments, K, and the array level. As mentioned previously, K equals the number of factors in the analysis, and the array level contains the limits of the number of categories of each factor. The other arguments required by the routines are either variables or arrays defined by the subroutines themselves or have been computed during the input portion of the program.

Meanq returns a complete analysis of variance based upon means. To produce a scaled estimate of error to be used with the values returned from Meanq, Hmean is divided into Sserr (Meyers, 1966). Fratios for main effects and interactions are then computed. As an alternative, one could scale the values returned by Meanq upward by multiplying each by Hmean and computing F ratios, using

the computed Sserr directly as the estimate of error (Winer, 1962).

The procedure outlined above has a number of features that should be emphasized. Initially, such a program can be used for equal, proportionate, and disproportionate cell sample sizes. Analysis of unweighted means for the equal and proportionate cases yields exact Fratios, so the E does not sacrifice accuracy by basing the analysis on means rather than on individual data points. In addition, the storage requirements are reduced dramatically from the situation where the analysis uses the individual data elements.

For example, the size of the array needed to accommodate the data points for a 3^5 factorial with 15 scores per cell, using the three subroutines mentioned earlier, is 16,384 locations, with each location requiring 4 bytes of core storage. In contrast, the procedure outlined above, using means, would require 1,024 locations for the cell means, plus 15 additional ones for the cell elements.

In general, analysis based on the individual data points will require N times more storage than an analysis based upon unweighted means. Such a difference in core storage requirements for two procedures that yield equivalent answers in two of three

situations is an important consideration. This is particularly the case for Es having access to small computers such as IBM's 1130 series. For the case where the cell sizes are disproportionate, the use of unweighted means is the E's only realistic option.

Finally, the computational scheme used in the subroutines in implementing Hartley's algorithm is based on deviations from cell totals. This procedure minimizes the likelihood that high-order digits will be lost in the subtract magnitude operation during intermediate computations. Consequently, the results returned by the program are expected to have good relative error.

REFERENCES

- HARTLEY, H. O. Analysis of variance. In A. Ralston and H. Wilf (Eds.), *Mathematical methods for digital computers*. New York: Wiley, 1967. Pp. 221-230.
- HEMMERLE, W. J. *Statistical computations on a digital computer*. Waltham: Ginn Blaisdell, 1967.
- IBM Scientific Subroutine Package. White Plains: IBM, 1969. Manual H20-0205-3.
- MEYERS, J. *Fundamentals of experimental design*. Boston: Allyn & Bacon, 1966.
- WINER, B. J. *Statistical principles in experimental design*. New York: McGraw-Hill, 1962.

Using a general analysis of variance algorithm for covariance designs

JOHN F. WALSH

Fordham University, Bronx, New York 10458

Procedures are described which enable researchers to implement balanced covariance designs of from one to four independent variables. Use is made of three subroutines from IBM's Scientific Subroutine Package which implement a general decomposition algorithm for balanced designs. FORTRAN instructions, illustrating the main calling program, are given.

Application of the analysis of covariance technique by behavioral scientists in research areas is extensive. The combination of interest in the adjustments provided plus the considerable computational requirements of the procedure have made this program one of the first to appear in computer-center libraries. Usually, however, library programs such as the biomedical series (BMD) from the University of California (Dixon, 1970) require large amounts of core and are not easily modifiable by the user. One runs them as is, or else does the analysis by hand.

The present article is intended for

researchers with small-to-modest computer configurations. The material below describes procedures which permit persons with modest knowledge of FORTRAN to implement covariance designs quite easily. Use is made of three subroutines—AVDAT, AVCAL, and MEANQ—from the Scientific Subroutine Package (SSP) (IBM, 1969). These routines implement the general decomposition algorithm developed by Hartley (1962).

Table 1 presents the FORTRAN code for the main calling program. The arrays are set to accommodate designs involving from one to four

independent variables with N Ss per cell. In determining the array sizes, space was provided for the inclusion of Ss as a pseudofactor. Specifically, the arrays are set to accommodate four independent variables plus the pseudofactor of Ss. The size of each of the arrays, M, for the covariate X, variate Y, and crossproduct variable XY, is computed as follows:

$$M = \prod_{i=1}^K [\text{LEVEL}(I) + 1],$$

where LEVEL(I) = number of levels of the factor and K = number of factors, including Ss.

On the first input card, the user punches the required parameters in I2 format. The parameters are the number of factors, K, and the number of categories or levels in each of them. The levels of the pseudofactor S are listed last. The FORTRAN variable names for the factors are II, JJ, LL, MM, and NS, respectively. If less than four independent variables are involved, the value of 1 is entered for those factors not included in the design. For example, if the design being analyzed involved a 3 by 4 analysis with 10 Ss per cell, the first

TABLE 1
FORTRAN CODE OF THE MAIN PROGRAM FOR ANALYSIS OF COVARIANCE

```

DIMENSION LEVEL(5), ISTEP(5), KOUNT(5), MSTEP(5), LASTS(5), SMX(31)
DIMENSION SMXY(31), SMY(31), NDF(31), NAME(15), SOX(31), SQY(31)
DIMENSION SQXY(31), X(800), Y(800), XY(800)
COMMON ERRORX, ERRORY, ERRXY, NDFERR, ADJERR, ADJSS(31)
DATA ERRORX, ERRORY, ERRXY, NDFERR/3*0., 0./, NAME/' A', ' B', ' AB',
1 ' C', ' AC', ' BC', ' ABC', ' D', ' AD', ' BD', ' ABD', ' CD',
2 ' ACD', ' BCD', ' ABCD'/
READ(1,2) K, I, J, L, M, NS
2 FORMAT( 6I2)
LEVEL(1)=I
LEVEL(2)=J
LEVEL(3)=NS
C SORT DATA ON SUBSCRIPTS ASSOCIATED WITH FACTORS: I-J-L-M-S
N=I+J+L+M+NS
DO 10 I=1, N
  READ(1,11) X(I), Y(I)
C ENTER FORMAT STATEMENT FOR DATA HERE
10 XY(I)=X(I)+Y(I)
  CALL AVDAT(K, LEVEL, N, X, L, ISTEP, KOUNT)
  CALL AVCAL(K, LEVEL, X, L, ISTEP, LASTS)
  CALL MEANO(K, LEVEL, X, GM, SOX, NDF, SMX, MSTEP, KOUNT, LASTS)
  CALL AVDAT(K, LEVEL, N, Y, L, ISTEP, KOUNT)
  CALL AVCAL(K, LEVEL, Y, L, ISTEP, LASTS)
  CALL MEANO(K, LEVEL, Y, GM, SOY, NDF, SMY, MSTEP, KOUNT, LASTS)
  CALL AVDAT(K, LEVEL, N, XY, L, ISTEP, KOUNT)
  CALL AVCAL(K, LEVEL, XY, L, ISTEP, LASTS)
  CALL MEANO(K, LEVEL, XY, GM, SOXY, NDF, SMXY, MSTEP, KOUNT, LASTS)
  NOUT=(2**K)-1
  DO 15 J=1, NOUT
15 SQXY(J)=(SOXY(J)-SOX(J)-SOY(J))/2.
    NSUBJ=2**K-1
    DO 20 J=NSUBJ, NOUT
      NDFERR=NDFERR+NDF(J)
      ERRORX=ERRORX+SOX(J)
      ERRORY=ERRORY+SOY(J)
20 ERRXY=ERRXY+SOXY(J)
      ADJERR=ERRORY-(ERRXY**ERRXY)/ERRORX
      NDFERR=NDFERR-1
      ERR=ADJERR/NDFERR
      NOUT=NSUBJ-1
      DO 25 J=1, NOUT
        CALL ADJUST(SOX(J), SOY(J), SOXY(J), J)
25 CONTINUE
      DO 30 J=1, NOUT
        SMSO=ADJSS(J)/NDF(J)
        F=SMSO/ERR
30 WRITE(3,31) NAME(J), NDF(J), ADJSS(J), SMSO, F
      WRITE(3,32) NDFERR, ADJERR, ERR
C ENTER OUTPUT FORMATS HERE
      CALL EXIT
      END
      SUBROUTINE ADJUST(A, B, AB, J)
      COMMON ERRORX, ERRORY, ERRXY, ADJERR, ADJSS(31)
      TX=A+ERRORX
      TY=B+ERRORY
      TXY=AB+ERRXY
      AA=TY-(TXY**TXY)/TX
      ADJSS(J)=AA-ADJERR
      RETURN
      END

```

12 columns would be punched as follows: 030304 010110.

Following the initial read statement, the first $K-1$ locations in the array LEVEL are assigned the values of the independent variables. NS is assigned to the K^{th} location in LEVEL. Examination of the statements in Table 1 should make the procedure clear. These arithmetic assignment statements are the only insertions made by the user. The addition or deletion of these assignment statements depends upon the number of factors involved. For designs with the same number of factors, no program change is required. All other parameters are computed within the program, using the user-supplied parameter, K.

To prepare the data cards for input, they are sorted, in turn, from the most rapidly changing index, I, to the least rapidly changing, S. For a 2 by 2 design with N Ss per cell, the sorting pattern would be: $A_1 B_1 S_1$, $A_2 B_1 S_1$, $A_1 B_2 S_1$, $A_2 B_2 S_1$, $A_1 B_1 S_2$, ..., $A_2 B_2 S_n$.

To incorporate Hartley's decomposition algorithm into covariance designs requires two steps. During input, one creates the dummy variable, $X(I) + Y(I)$. In the program, it has the variable name XY. The array of sums, XY, is passed to the three subroutines AVDAT, AVCAL, and MEANO, just as are the arrays for the covariate X and the variate Y. Recovery of the crossproduct variation, $X * Y$, is obtained by using

the identity,
 $XY = [(X + Y)^2 - X^2 - Y^2]/2$
 (Hemmerle, 1967). This latter step is illustrated by the FORTRAN instructions contained in the DO 15 loop in Table 1.

The complete factorial decomposition of the variables X, Y, and $X * Y$, are contained in the arrays SQX, SQY, and SQXY, respectively. The computation of the adjusted sums of squares required for the covariance analysis is performed in the subroutine ADJUST. This subroutine is included in Table 1. The sum of squares and degrees of freedom associated with the S factor and the interactions of this factor with the $K-1$ independent variables are accumulated in the DO 20 loop. These accumulated values provide the within-cells estimate of error, and the degrees of freedom for error.

The instructions in the DO 30 loop compute the F ratios and output the results. The labeling of the factors on output is controlled by the array NAME, which is initialized by a DATA statement. The variable associated with the most rapidly changing subscript, I, is labeled A, the next most rapidly changing variable has the subscript J and is labeled B. This pattern is followed with the other variables, subscripts, and labels.

The code used in Table 1 provides the opportunity for the researcher to have a "hands on" sense about the analysis. The instructions that the researcher would desire to change are visible and not extensive. These features are attractive to the person who wants to insert other or additional labels as part of the main program. In contrast, the sheer volume of the code involved in library programs such as the BMD (Dixon, 1970) inhibits tinkering.

REFERENCES

- DIXON, W. (Ed.) *Biomedical computer programs. University of California publications in automatic computation*. No. 2. Berkeley: University of California Press, 1970.
- HARTLEY, H. O. Analysis of variance. In A. Ralston and H. Wilf (Eds.), *Mathematical methods for digital computers*. (2nd ed.) New York: Wiley, 1962. Pp. 221-230.
- HEMMERLE, W. J. *Statistical computations on a digital computer*. Waltham: Ginn Blaidell, 1967.
- International Business Machines Corp. Scientific Subroutine Package. Manual H20-0205-3. White Plains, N.Y: IBM, 1969.