```
*        LABEL
*        FAP
* SSH              STORE-TO-STORE HOLLERITH
         COUNT     22
         ENTRY     (CSH)
         ENTRY     (SCH)
(CSH)    LDQ       NOP
A        CLA       AC
         TRA*      $ (IOH)
NOP      NOP       * + 1
         TRA       1, 4              DUMMY READ
(SCH)    LDQ       TRA
         TRA       A
TRA      TRA       * + 1
         PXA       0, 4              DUMMY PUNCH
         COM
         ADD       A3
         STA       X
         ADD       A5
         STA       T
X        XEC       **
T        TRA       **
A3       PZE       3
A5       PZE       5
AC       PZE       ,,2
         END
```

Examples

(a) To spread a packed array.

```
      PUNCH 100, (DICT (I), I = 1, 20)
100   FORMAT (20A6)
      READ 101, (EXPND (I), I = 1, 120)
101   FORMAT (120A1)
```

(b) To pack a symbol of up to six characters, left adjusted, filled out with blanks.

```
      PUNCH 200, (CH(I), I = 1, NCH)
200   FORMAT (6A1)
      READ 201, SYMB
201   FORMAT (A6)
```

(c) To convert an array of characters to corresponding integers.

```
      PUNCH 300 (IBFR (K), K = 1, 20)
300   FORMAT (20 (2H00, A1, 3H000))
      READ 301 (IBFR (K), K = 1, 20)
301   FORMAT (20A6)
```

(d) To prepare a format statement $(In, 2X)$ where n is not known in advance.

```
      PUNCH 400, N
400   FORMAT (2H(I, I2, 4H, 2X))
      READ 401, FT
401   FORMAT (12A6)
```

(e) To determine nearest integer multiple of a power of 10 to a general floating-point number.

```
      PUNCH 310, X
310   FORMAT (1PE 10.0)
      READ 311, Y, I
311   FORMAT (F 5.0, 1X, I3)
      Z = Y * 10. ** I
```

(f) To read in data as specified by code letters on the cards, to allow more flexible input preparation.

```
      READ IN CODE LETTERS
      READ INPUT TAPE 2, 200, (HX (I), I = 1, 6)
200   FORMAT (6A1)
1     READ INPUT TAPE 2, 200, H
      DO 3 J = 1, 6
      IF (H - HX(J)) 3, 2, 3
2     GO TO (11, 12, 13, 14, 15, 16), J
3     CONTINUE
      CALL DIAG
11    READ 201, (TITLE (I), I = 1, 11)
201   FORMAT (6X, 11A6)
      WRITE OUTPUT TAPE 3, 201, (TITLE (I), I = 1, 11)
      GO TO 1
12    READ 202, I1, I2, I3
202   FORMAT (6X, 11I6)
13    READ 201, (FORM (I), I = 1, 11)
      READ FORM, A, B, C
      GO TO 1
14    READ 201, (FOUT (I), I = 1, 11)
      GO TO 1
15    READ 205, X, Y, Z
205   FORMAT (6X, 6F 10.0)
      GO TO 1
16    CONTINUE
```

I. C. Pyle
A.E.R.E.
Harwell, England

A COMPUTER TECHNIQUE FOR HANDLING ANALYSIS OF VARIANCE

Introduction. In writing a completely general one-variate analysis-of-variance program for a digital computer, one is confronted with the problems of (1) devising a systematic scheme for retrieving from the computer storage the elements that occur in each of the many sums necessary for the analysis desired and (2) formulating and solving a set of least squares equations. The purpose here is to show how subscripts can be used in a computer program to answer these two problems and to control all the computations required. We use the method of solving a set of equations for computing the adjusted sums of squares required.

The Method. For ease of discussion, let us consider data that can be classified by three components, say A, B, and C, with a levels in A, b levels in B, and c levels in C, and which has n_m observations in the mth cell.

We sum this data over each cell and record (store in a computer storage) the summarized data and the subscript array for each cell along with the number of observations comprising each sum, as in Table 1.

Suppose that we wish to compute the sums of squares for each so-called "main effect" and all possible interactions by forming and solving the appropriate set of least squares equations. This set of equations is completely specified by the subscripts $i\cdot\cdot$, $\cdot j\cdot$, $\cdot\cdot k$, $ij\cdot$, $i\cdot k$, $\cdot jk$, ijk; $i = 1,2, \ldots, a$; $j = 1, 2, \ldots, b$; $k = 1, 2, \ldots, c$ which we shall consider to be row headings and column headings for the matrix of coefficients.

It is convenient at this point to accumulate and store a sum of observations from Table 1 for every subscript which appears as a row heading for the coefficient matrix. To do this we add together those $X_{ijk\cdot}$ from Table 1 whose subscripts are equal to a particular row heading subscript in corresponding nondot positions. Denote this array of sums by the vector S.

TABLE 1. Summarized data, subscripts and frequencies

Cell No.	$X_{ijk\cdot}$	*Subscript*	*Frequency*
1	$X_{111\cdot}$	111	n_1
2	$X_{112\cdot}$	112	n_2
⋮	⋮	⋮	⋮
c	$X_{11c\cdot}$	$11c$	n_c
$c + 1$	$X_{121\cdot}$	121	n_{c+1}
⋮	⋮	⋮	⋮
$(a)(b)(c)$	$X_{abc\cdot}$	abc	$n_{(a)(b)(c)}$

Similarly, we can accumulate and store the frequencies from Table 1 for each row heading. These numbers are then the elements of the diagonal of the coefficient matrix. Denote this array by the vector D. The matrix vector of constants of the set of equations we wish to solve is $S - MD$ where M is the overall or grand mean.

(*Continued on page 449*)

PRACNIQUES, HOWELL—Cont'd from page 433

There remains yet the task of forming the off-diagonal elements of the coefficient matrix. To do this, we apply the following rule which is stated here in a way that can be easily programmed for a computer.

Rule. Compare the row and column headings of each element L of the coefficient matrix.

(1) If the row and column headings are alike, then the desired element is obtained from the vector D.

(2) If there is at least one summation dot in each corresponding position (i, j and k) of the row and column headings of an element L, then a new subscript is formed by adding these two headings, treating the summation dots as zeros. This new subscript is used as before to accumulate the desired element from the frequencies in Table 1.

(3) If neither (1) nor (2) holds, then $L = 0$.

In case the computer storage is not large enough to store this matrix then the set of equations can be solved by the method of iteration and each element of the matrix can be generated as needed, used and then discarded. Once the solution to this set of equations is obtained, the sums of squares desired can be computed in the usual way.

Conclusion. This method of using subscripts to control all the calculations required for an analysis of variance has been used by the author in programs written in Soap language for the basic IBM 650 computer and in Fortran language for the IBM 7090 computer. Both programs are efficient and are in current use.

Acknowledgment. The author wishes to thank Mark Robinson for suggestions concerning the writing of this information, and Dr. A. E. Brandt for suggesting work that led to the original Type 650 computer prgram.

Bibliography:

1. Brandt, A. E. The analysis of variance in a $2 \times s$ table with disproportionate frequencies. *J. Amer. Stat. Assoc. 28*, 182 (June 1933).
2. Yates, F. The analysis of multiple classification tables with unequal numbers in the different classes. *J. Amer. Stat. Assoc. 29*, 185 (Mar. 1934).
3. Snedecor, George W., and Cox, Gertrude M. Disproportionate subclass numbers in tables of multiple classification. Iowa State Agricultural Experiment Station Bull. No. 180, Ames, Iowa (1935).
4. Seeley, Burton D. An approach to problems involving disproportionate frequencies. Presented at the Philadelphia meeting of the Institute of Mathematical Statistics, Dec. 27, 1939.
5. Wald, A. On the analysis of variance in case of multiple classifications with unequal class frequencies. *Ann. Math. Stat., 12* (1941), 346–349.
6. Hazel, L. N. The covariance analysis of multiple classification tables with unequal subclass numbers. *Biometrics Bull. 2*, 2 (Apr. 1946).
7. Stevens, W. L. Statistical analysis of a non-orthogonal trifactorial experiment. *Biometrics 35* (1948), 346–367.
8. Crump, S. Lee. The present status of variance component analysis. *Biometrics. 7*, 1 (Mar. 1951).
9. Henderson, C. R. Estimation of variance and covariance components. *Biometrics. 9*, 2 (June, 1953).
10. Harvey, Walter R. The Analysis of Non-Orthogonal Data With IBM Equipment. *IBM Tech. Newsletter 9* (Jan. 1955).
11. Snedecor, George W. *Statistical Methods.* Ames, Iowa: The Iowa State College Press, Fifth Ed., 1956.
12. Hartley, H. O. Programming analysis of variance for general purpose computers. *Biometrics 12*, 2 (June 1956), 110–122.
13. Kunz, Kaiser S. *Numerical Analysis.* McGraw-Hill Book Co., Inc., 1957.

Research Summaries

PHILIP BAGLEY, MANDALAY GREMS, **Editors**

A Generalized Combinatorial Problem

Hughes Aircraft Co., Fullerton, Calif.

Reported by: John G. Haynes (June, 1962)

Descriptors: **graph theory, combinatorial analysis, organization of information**

The objective of this research was to establish the following two theorems:

I. *Every connected graph is of some definite order-k.*

II. *Every planar connected graph is at most of order-4.*

Order-k is defined: A connected graph is said to be a *connected graph of order-k* if and only if at least one arrangement of the points into k-subsets is possible and no arrangement into less than k-subsets is possible, each such subset being of connectivity-zero.

The method of approach was:

1. To provide a foundation for pairs of subsets of connectivity-zero.
2. To derive the class of all basic graphs of order-k.
3. To show that every graph of order-k contains at least one basic graph of order-k.
4. To show that all basic graphs of $k \leqq 4$ are planar.
5. To show that all basic graphs of $k > 4$ are non-planar.

The major results of this research are:

1. Provides a foundation for connected graphs which differs from classical treatment.
2. Provides a basis for a proof of the four color graph theorem.

The following appear to be among the areas of immediate application:

1. Electrical network analysis
2. Binary matrix analysis
3. Flow diagram structure and analysis
4. Complex analysis of organization of information.

---

14. Scarborough, James B. *Numerical Mathematical Analysis.* Baltimore: The Johns Hopkins Press, Fourth Ed., 1958, pp. 554–558.
15. Harvey, Walter R. Analysis of data with unequal numbers where a set of orthogonal comparisons among the subclass means is desired. Paper presented in the Biometrics Society Section of the AIBS meetings at Pennsylvania State University, Aug. 31, 1959.
16. Lieblein, Julius. A general analysis of variance scheme applicable to a computer with a very large memory. *J. ACM 6*, 4 (Oct. 1959), 469–475.
17. Scheffe, Henry. *The Analysis of Variance.* John Wiley & Sons, Inc. New York, 1959.
18. *The Analysis of Variance. Biometrics 3*, 1 (Mar. 1947) (special issue).
19. *Components of Variance. Biometrics 7*, 1 (Mar. 1951) (special issue).
20. *The Analysis of Covariance. Biometrics 13*, 3 (Sept. 1957) (special issue).

John R. Howell*
The Martin Co.
Orlando, Fla.

* Now at Lockheed-Georgia Co., Marietta, Ga.