Tabulation of Vital Records by Computer

A. M. GITTELSOHN, Ph.D., and R. S. SENNING, B.S.

THE FREQUENCY tabulation is one of the most widely used and best understood means for summarizing data. As such, the development of methods for preparing tabulations is of general interest. The advent of high-speed digital computers together with peripheral data storage media has opened up a new and broadened range of processing possibilities to be considered along with manual and mechanical systems. The economic method for tabulation and summarization of a given file of data will depend on at least three intrinsic factors: total number of observations, size and complexity of each observation, and form of the output tables. When any one of the three is large, the conditions may be appropriate for computer application. The present objective is to outline approaches to mass data tabulation and other associated problems being undertaken in the office of biostatistics of the New York State Department of Health.

Formulation

At the outset, we shall require a common terminology and notation to describe the generation of cross-classifications. Where possible, formulas are shown as they would be coded in an algorithmic language. The symbol (*) in an algebraic context denotes the operation of multiplication. A frequency table is a matrix with dimensionality equal to the number of variables being cross-classified and with number of elements (frequencies) equal to the product of the number of categories or levels in each dimension. Thus, a multiple classification of people by sex, marital status, and age groups is three-dimensional. If there are 4 marital states and 10 ages, the number of cells in the table will be $2\times4\times10=80$ classes in all. To reference a given cell in the table, say T(I,J,K), it is necessary to specify the level of each variable. Letting "male" be the first level of sex, "married" the second level of "marital status," and "40–49" the fifth level of age, T(1,2,5) will be the location of the cell in the array in which the married men aged 40 to 49 years will be counted.

It is generally of interest to examine marginal frequencies and subtotals in multiple classifications. An N-way table contains the requisite information for all cross-classifications of lower dimensionality than N among the same set of variables. For the three-variate case of sex, marital status, and age, there are three two-dimensional and three one-dimensional sets of marginal frequencies. The grand total may be regarded as a single zero-dimensional table. Hence, eight arrays are required to describe all cross-classifications among the three variables taken 0, 1, 2, and 3 at a time.

The total of 165 cells required for all classifications among the three variables is the product of the number of levels plus one for the total in each dimension: $165=(2+1)\times(4+1)\times(10+1)$.

To arrive at a compact notation, this suggests assigning the level 1 to the total over each dimension. Letting the subscripts I, J, and K index sex, marital status, and age respectively, the single table T(I,J,K) will suffice to represent all data configurations. T(1,J,K) refers to the Jth level of marital status and the Kth level of age over both sexes; T(I, 1, 1) refers to the Ith

Dr. Gittelsohn is director of health statistics and Mr. Senning is senior biostatistician, office of biostatistics, New York State Department of Health. *This study was supported in part by Public Health Service grant CH 00080.*

level of sex over all marital states and ages; and T(1, 1, 1) refers to the grand total as shown below.

Table No.	*Dimensions*	*Variables*	*Cells (N=165)*	*Matrix*
1	3	sex × MS × age.	80	T(I, J, K)
2	2	sex × MS	8	T(I, J, 1)
3	2	sex × age	20	T(I, 1, K)
4	2	MS × age	40	T(1, J, K)
5	1	sex	2	T(I, 1, 1)
6	1	MS	4	T(1, J, 1)
7	1	age	10	T(1, 1, K)
8	0	grand total	1	T(1, 1, 1)

In general, consider a cross-classification of N variables with V(I) representing the Ith variable, I=1,2, . . ., N. Let L(I) be the number of mutually exclusive categories, subtotals, and grand total, in the Ith dimension. The total number of cells in the matrix will be the product

$$\text{TOTAL}=L(1)*L(2)*\ldots*L(N). \qquad [1]$$

Each cell in the matrix will be addressed by specifing a value of the vector, [V(1), V(2), . . . ,V(N)]. The number of K-dimensional marginal tables contained in the setup is the number of ways K subscripts can be chosen out of N, or the combination of N elements taken K at a time, N!/K!(N—K)!. Summing over K, the total number of marginal frequency tables plus the N-way table is

$$\sum_{K=0}^{N} \frac{N!}{(N-K)!K!}=2^N.$$

The memory of computers is one-dimensional in that core locations are addressed by the integer sequence 0,1,2,3, ... Hence, a multiple subscripting scheme must be reduced to a single subscript either explicitly or implicitly in the course of representing a data array in memory. The problem is to express the vector [V(1), V(2), . . . , V(N)] as a dense integer set with range equal to the total number of cells in the table. Each V(I), as defined, has L(I) levels referenced by the ordinal numbers 1,2, . . . , L(I). For any given vector these may be transformed to a single subscript J with the desired ordinal property by defining for say N=4.

$$J=V(1)+L(1)*[V(2)-1+L(2)*[V(3)-1+L(3)*[V(4)-1]]]. \qquad [2]$$

Letting the counting modulus for variable I be defined as the product

$$\text{MOD}(I)=L(1)*L(2)*\ldots*L(I-1). \qquad [3]$$

The relation may be more simply expressed as the sum

$$J=1+\sum \text{MOD}(I)*[V(I)-1] \qquad [4]$$

where MOD(1)=1.

Consider the two representations of the two-way classification of persons by sex and age as shown in table 1. The number of levels, L(I), in the first dimension is three and, in the second, five. The counting moduli are MOD(1)=1 and MOD(2)=3. By equation (4), the relation between J and the V(I) is

$$J=V(1)+3*[V(2)-1].$$

By (1), the total number of cells in the table is the product of the number of levels in each dimension, or could be redefined by (3) as TOTAL=MOD(N+1)=3×5 and the index J has range 1, 2, . . ., 15. In an N-way classification, the computation of J by equation (2) involves (N-1) multiplications. Since a multiply by computer may require up to 10 times as long to execute as an addition, considerable operating efficiency will be gained if the problem is approached from the latter standpoint. This can be easily accomplished if the variables are initially recoded by the program in terms of the counting moduli. For the sex-marital status table, the following set of codes would suffice:

Variable	*Code*
Sex	1 male
	2 female
Marital status	3 single
	6 married
	9 widowed
	12 divorced

Letting [W(I)] represent the Ith variable coded under such a scheme, equation (2) for this example reduces to the sum J=1+W(1)+W(2) and in general to

$$J=1+W(1)+W(2)+\ldots+W(N). \qquad [5]$$

Table 1. Two representations of the two-way classification of persons by sex and age

Marital status	(1) Total	(2) M	(3) F
	Two subscripts [V(1), V(2)]		
(1) Total	1, 1	1, 2	1, 3
(2) Single	2, 1	2, 2	2, 3
(3) Married	3, 1	3, 2	3, 3
(4) Widowed	4, 1	4, 2	4, 3
(5) Divorced	5, 1	5, 2	5, 3
	One subscript [J]		
(1) Total	1	2	3
(2) Single	4	5	6
(3) Married	7	8	9
(4) Widowed	10	11	12
(5) Divorced	13	14	15

Computation of J by equation (5) rather than equation (2) will result in an execution time-saving of up to 80 percent.

To carry out a tabulation problem we begin by defining one or more counting arrays in storage. The tabling procedure consists in successively reading for each case the value of the label vector [W(I)], which represents the levels of the variables of interest computing the index J, adding one to the frequency contained in T(J), and continuing with the next case until the entire file has been examined. Subtotals and marginal totals are obtained by summing frequencies from the N-dimensional matrix into the tables of lower dimensionality. The final printed output simply may be the cell frequencies set out in tabular form whereby rows, columns, hyper-rows, hyper-columns, and pages are labeled for convenient readability. Subsequent operations may include norming for percentages or rates and statistical manipulation appropriate to the particular problem. Of particular pertinence in the context of categorical data are chi-square analyses, life tables, and discrete stochastic processes.

Encoding

Tabulation may be viewed as a problem of encoding. For each element being cross-classified, we begin with a label vector $[C_1C_2 \ldots, C_s]$ where each C_k is an alpha-numeric character. The essence of the problem is to define a series of efficient transformations on the C_k which will lead to the desired single subscript J. A code is a correspondence between the categories of a classification and a character set or alphabet. In general, code systems are not ordinal which is the property we seek for the index J.

Scale or translation, or both, transformations of the type $W_k = a + bC_k$ may sometimes be employed. Thus, a recoding of AGE in single years to 5-year age groups may be effected by setting $W = 1 + AGE/5$, the divide being in fixed point (truncation to right of decimal point). An alternate would be to use AGE as the argument of the table R where

$$R(0) = R(1) = \ldots = R(4) = 1$$
$$R(5) = R(6) = \ldots = R(9) = 2$$
$$\vdots$$
$$R(95) = R(96) = \ldots = R(99) = 20.$$

The recode could now be done with table lookup by setting W=R(AGE). If the counting modulus for age was something other than 1, this would be reflected in the tables. In general, the table lookup will be up to 5 times more efficient than the multiply or divide. For categorical variables where the code assignments are purely arbitrary, the use of recode functions may be the only alternative for computing W.

The direct use of a recode table R for a four-digit variable, such as the World Health Organization classification of cause of death, requires 10,000 cells for the table itself. To conserve storage, an alternative might be to define two variables for cause

A = first three digits
B = fourth digit

and to let the coded value of cause be computed by

$$W = R(A) + B.$$

In this case, the storage requirement for R would be 1,000. Further space savings may be realized by combining branching with table lookups or by using multiple lookups of the form

$$W = R1(R2(R3(X) + Y) + Z).$$

An additional alternative includes sorting, which imposes the requirement for multiple reading of the data file.

The steps in tabulating a file of cases may be summarized as follows:

1. Set up recode tables and counting arrays.
2. Tabulation:
   a. Read label vector for one case
      [CODE(S)]
   b. Encode label vector to single subscript J
      J=R[CODE(S)]
   c. Add 1 to appropriate cell of frequency array by replacing T(J) with T(J)+1. Execute steps a-c for each successive case until file has been exhausted.
3. Compute marginal frequencies by summing over disjoint categories.
4. Compute statistical indexes, estimates, and test statistics, and write tabular output for printing.

The characteristics of the data file to be processed, the form of the output data configurations, and the type of equipment available dictate the possible avenues of approach to development of cross-classifications. The critical aspects of the input file and the output tables are the volume of cases, the form of the code vector $[C_i]$ from which J is to be computed, and the maximum value in the range of J. Important equipment considerations include character transmission rates of input and output devices, cycle time, intermediate storage, and the size of internal storage.

By using an N-dimensional array to encompass several (N–K) dimensional arrays, a single addition will suffice for each case being tabulated. The tables of lower dimensionality than N may be obtained by summing frequencies. This procedure will constitute an advantage when the expected frequency per disjoint cell is greater than one. By allowing for totals and subtotals within a given dimension, a notational symmetry is gained at the expense of increasing the size of the counting array. The advantage here lies in the possibility of developing generalized summation and output routines. Clearly, many approaches to tabulation are admissible, each presenting advantages and disadvantages. The appropriate approach will depend on achieving a balance between equipment, programing, and tabulation problem considerations.

Example

Suppose the problem is one of preparing a four-dimensional table and combinations of k-dimensional marginal tables among four variables, cause, age at time of death, race, and sex, the data file consisting of 200,000 death records. Let the variables after recoding be:

		Levels		
I	*Variable*	*L(I)*	*MOD(I)*	*Groupings*
1	Cause	100	1	Total and 99 cause groups.
2	Age at death.	20	100	Total and 19 five-year age groups.
3	Race	3	2,000	Total, white and nonwhite.
4	Sex	3	6,000	Total, male and female.

The total number of cells in the table, including all marginals, is the product of the number of levels for each variable, that is, $3\times3\times20\times100=18{,}000$. Since each cell is to be a frequency, 18,000 units of storage to serve as counters are required to complete the tabulation. This far exceeds the counter capacity of mechanical equipment and the storage of most commercial or data-processing type computers. Under such circumstances, the usual approach is to reduce the problem to a series of subproblems by rearranging the case file in a given sequence, say by cause of death. Each cause subset of the ordered file is treated, in effect, as a separate tabulation input. The sorting process, preliminary to the counting, may require the major portion of the time and cost of the entire job. Large-memory computers may be capable of handling the problem without sorting and with a single input of the data file.

With limited memory, a second possibility would be to use intermediate storage. The number of disjoint classes in the frequency array is

$$7{,}524=(3-1)\times(3-1)\times(20-1)\times(100-1).$$

Given sufficient internal storage, the 7,524 cell four-dimensional table without marginal totals

can be produced at initial input. Through a secondary input of the table, the storage can be re-used to obtain the subtotal classes by appropriate logical and summation operations.

Multiple passes of the input file can be used to reduce further the internal storage requirement. For example, males can be processed at the first input and females at the second, or whites at the first and nonwhites at the second. An additional possibility arises when it is recognized that the maximum frequency in any cell will be 200,000, the total number of deaths in the file. An integer as large as 260,000 can be handled by 18 binary bits. Hence, when dealing with binary computers of word size 36 bits or greater, the word can be divided into right and left halves. For binomial traits such as sex, males can be counted in the right portion and females in the left, thus doubling the effective size of the computer memory. In this case, the procedure would be to determine the index J as a function of the three variables, cause, age, and race; if the death were a male, to add 1 to counter T(J) or, if female, to add 2^{18} to counter T(J). In such a circumstance, the trade-off would be between multiple input and the masking and shifting operations necessary to deal with parts of words during the computation and output phase.

The emphasis in the office of biostatistics has been on tabulation problems where internal storage is not a restriction or where intermediate storage can be effectively used. By concentrating on large-memory binary computers which are available in most large cities, the situation has rarely arisen where the total number of frequencies exceeds available storage. With a 32,000 word memory and a 48-bit word, a tabulation array with more than 50,000 cells can be developed, allowing for a maximum frequency of over 16 million per cell. Assuming that 500 frequencies (20 columns and 25 rows) with titles and labels can be fitted onto a printed page, the 50,000 cell tabulation will require at least 100 pages.

In setting up a series of tabulations, it is evident that some selectivity must be exercised by way of achieving a balance between sample size and table size. Suppose a questionnaire consisting of N=15 yes-no items is administered to 32 subjects. The total number of k-dimensional tables (k=0, 1, . . ., 15) which can be developed from the file is $2^{15}=32{,}768$. The single 15-dimensional table will contain a total of 2^{15} cells and the expected frequency per cell in the table is 0.001. As the number of variables and the number of levels per variable increase linearly, the possible output configurations increase geometrically. In a typical mortality classification system with a total of 20 variables and a geometric mean of 25 levels per variable (say 20 two-digit variables), there are more than 1 million tables of dimensionality up to 20 and 25^{20} cells in the single 20-dimensional table.

Programing

During the past 2 years, members of the staff of the office of biostatistics have been exploring the feasibility of using large memory binary computers to produce mortality and natality tabulations from vital records. A central theme has been the development of frequency matrices using random sequenced files as input, thereby avoiding the necessity for preliminary sorting.

Initially, the programing function presented major difficulties in that no computer technologist positions exist in the office and assistance had to be obtained through outside sources. It has become evident that programing in algorithmic languages by statisticians is a practical solution which allows the statistician to continue to function in that capacity without requiring full-time professional programing support. A gratifying aspect of such an approach is that the object programs thereby produced are in a tolerable range of efficiency, and the programing effort does not require an unreasonable amount of time. It has become evident that the major factor affecting operating speed is not the source of language but the general formulation. A statistician, with an intimate knowledge of the data and the problem at hand, has a certain advantage over a programer unfamiliar with either.

At the present, the programing effort of the office is being carried internally. Program analysts are consulted occasionally concerning specific technical questions. A variety of programing problems have been attacked, ranging from tabulation of massive fixed-format data

files to data processing to mathematical-statistical analysis. The objective is to develop methodology and programing skills to the point where the time lag, between formulation of a question by a user and production of numerical results, is minimized.

Files Development and Maintenance

The setting up and maintenance of suitable input files of birth and death certificates are necessary initial steps in the tabulation of vital record data. Under a plan to preserve such information, magnetic tape files have been produced for births from 1955 and deaths from 1949 occurring in New York State, exclusive of New York City. Special files have been created for children born or dying, or both, with congenital malformations beginning with the birth year 1945 and for twin sets beginning with the year 1950. Currently, the annual volumes of births and deaths in the jurisdiction are 200,000 and 100,000 respectively.

Direct posting of data by case and digital coding is highly inefficient from the standpoint of space utilization of storage media. With tabulating cards, one card column is usually assigned to one digit of information. For a binomial variable such as sex, this leads to the assignment of only two of the 4,096 possible punch configurations in the column. In the binary coded decimal system (BCD), six binary positions are assigned to a one-character alphanumeric field. For sex, a single binary bit suffices to characterize the two possibilities of male or female, thus allowing a 6-to-1 space reduction relative to BCD coding. The variable "day of month" with 31 possibilities requires two BCD characters of 6 bits each, but only 5 bits with binary coding. In general, a code with upper limit M will require D bits in BCD coding and B bits in binary coding where

$$10^{D-1} \leq M < 10^{D}$$
$$2^{B-1} \leq M < 2^{B}.$$

The maximum saving occurs when recoding two-level characteristics. With 36-bit word computers, a total of 3 or 4 words constituting a logical record is more than sufficient to characterize the statistical information contained on a birth or a death certificate under current code systems. By blocking logical records into physical records of convenient length, say 256 words per block, tape files have been created wherein a single tape reel of 2,400 feet contains more than 300,000 cases, even when the tape density is as low as 200 characters per inch. With a higher blocking factor and high density tape (800BPI), over 1 million birth records can be stored on a single reel.

Tape formats for packed, binary coded birth and death records are shown in tables 2 and 3. Bits within words, numbered 1 through 36, have been assigned to the various characteristics. The basic operations for abstracting data from a given word are logical bit intersection and right shift. By not using the sign bit (bit number 1), characteristics stored in the extreme left portion of the word can be abstracted with a right shift only. Conversely, variables stored in the extreme right portion can be abstracted with a logical intersection only. By assigning variables often used in conjunction with each other in adjacent bit positions, a new combined variable can be defined with a single operation. Thus, the compound variable marital status-race-sex can be extracted from the first word of the record, and the index J for a three-way

Table 2. Data format for death record (packed binary tape), New York State, 1949–63

Card column	Item	Storage	
		WORD	BITS
	Place of occurrence:		
1–2	County	1	2–7
3–4	District	1	8–14
	Place of residence:		
5–6	County	1	15–20
7–8	District	1	21–27
62	Autopsy	1	28–29
47	Marital status	1	30–32
51	Race	1	33–35
46	Sex	1	36
9–14	Death certificate number	2	2–20
16	Institution of death	2	21–24
	Age:		
48–49	Number of units	2	27–33
50	Type of units	2	34–36
	Date of death:		
42	Month	3	2–5
43–44	Day	3	6–10
45	Year	3	11–14
	Cause of death:		
55	Fourth digit	3	23–26
52–54	First three digits	3	27–36

Table 3. Data format for birth record (packed binary tape), New York State, 1955–63

Card column	Item	Storage	
		WORD	BITS
1–4	Place of occurrence	1	2–14
5–8	Place of residence	1	15–27
15–16	Institution of birth	1	28–33
43	Plurality	1	34–36
9–14	Birth certificate number	2	2–20
69–70	Malformation and birth injury.	2	21–27
59	Weight at birth	2	29–32
56	Race	2	33–35
42	Sex	2	36
44–46	Date of birth, month and day.	3	2–10
64–67	Complications of pregnancy, labor, and delivery; operations.	3	11–23
48–49	Age of father	3	24–29
54–55	Age of mother	3	31–36
	Cause of death	4	1–10
57–58	Length of gestation	4	11–16
	Prior births:		
63	Born alive, now dead	4	17–19
62	Stillborn	4	20–22
60–61	Total	4	23–26
	Age at death:		
	Number of units	4	27–33
	Type of units	4	34–36

classification of deaths by sex, by race, and by marital status, can be computed with a single intersection and one table lookup R. Letting BOOL(WORD*MASK) represent the algorithmic expression for the operation of bit intersection on the two arguments WORD and MASK, where the qualifier "BOOL" indicates the symbol (*) is a logical operator, the statement for obtaining J would be of the form

$$J=R[BOOL(WORD(1)*177)].$$

The major advantage of using binary coded data as input into binary computers is that decimal or BCD to binary conversion is done only once. The advantage increases with the number of times the file is used. The placing of several characteristics in one word results in a space gain at the expense of requiring extra operations to manipulate the data. A major disadvantage of binary coded records is that such a file cannot be effectively processed by most commercial type computers.

Updating and file maintenance present no special difficulties when dealing with binary records. If the file is sequenced by birth certificate or death certificate number, the standard data-processing approach of merging applies. A current project is underway to update the birth file with matching infant and childhood mortality data. For those years which are not sequenced, if data-processing techniques are to be used, the approach would be to sort the file of births, sort the file of deaths, and then merge the two files. Such a procedure is indicated when processing is done on equipment with limited internal storage. An alternative approach on large memory binary computers would be to use all available storage for an updating table called U(K). Suppose the new data to be added to the birth record (age at death) can be characterized with four binary bits which allow for 16 code possibilities. Assuming a 36-bit word, each word in the array can store the updating data for 8 births, and a 25,000-cell array can store the mortality data for 200,000 births, the total number of births occurring within any one year. Assuming that each birth within a year is assigned a unique certificate number starting with 0 and ascending by ones, the storage pattern would be to assign the mortality data for the first eight birth certificate numbers 0 through 7 to successive 4-bit modules in U(1); for the second eight numbers 8 through 15 in U(2); and so on up to the last set of eight. Then each birth record would be read and its certificate number K abstracted from the record. The appropriate word J in the updating array U is determined by the fixed point divide: J=K/8. The 4-bit character corresponding to the Kth birth certificate is located within U(J) by using the remainder of the above division or the residual Z=K mod(8). Since the probability of dying in the first 5 years of life for New York births is less than 3 percent, more than 97 percent of the 4-bit characters will have the value O, corresponding to survival.

If, after it has been read into memory, the certificate number of a birth record to be updated occupies the first 20 bits of a 36-bit word, called WORD the sequence of programing steps might be written as:

1. Abstract the certificate number of the birth record with a right shift of 16 places.

   K=BOOL(WORD,R16)

2. Determine the residual Z=K mod(8)

   Z=BOOL(K*7)

   where 7 is the octal (base 8) representation of the binary mask 111.

3. Determine the location in the updating array where the information is stored by dividing K by 8 (three place right shift).

   J=BOOL(K,R3)

4. Branch on Z to the appropriate 4-bit character of U(J) and abstract this character C

   C=BOOL(U(J)*MASK(Z),R(Z)).

5. Add C to the birth record if nonzero.

Under such a setup, the entire updating procedure can be handled in a single pass of the file of 200,000 birth records. Compared with the tape sort and merge approach, a timesaving of up to 90 percent can be realized. A simple merge on two presequenced files will be only slightly more efficient. For any given updating problem the word size, the number of words available, the number of binary bits to be added, and the number of records in the file to be updated will determine the number of tape passes. With a 32,000 word machine having 48-bit words, 12 bits of information can be added to more than 100,000 records, reading the file once and only once.

Subsampling

A general problem that arises with manipulation of vital record files is the selection of subsets of birth or death records with specific characteristics. A recent example involved the combination of three separate requests for mortality data on a case basis: deaths under 15 years of age for residents of county X occurring outside the county, deaths from late effects of encephalitis, and deaths from lung cancer occurring to residents of county Y. Using a Boolean approach, variables were defined with respect to given attributes as follows:

Variable	*Attribute*
A	Age under 15
B	Residence in county X
C	Nonoccurrence in county X
D	Residence in county Y
E	Death from late effects of encephalitis
F	Death from lung cancer

The entire selection process involved abstracting the age, residence county, occurrence county, and cause of death from each death record, determining the binary values of A through F (1 if the case had the attribute, 0 if it did not) and testing the truth-value of the proposition

P=BOOL(A*B*C+E+D*F)

where the symbols (*) and (+) represent the logical AND and the logical OR respectively. The values of the variables, 1 or 0, act as truth values, true or false, respectively. If P=1 (true), the record was written out; if P=0 (false), the record was skipped. The use of the logical operators of intersection and union in such a context both simplifies program writing and leads to efficient operating programs. In general terms, binary computers lend themselves directly to such an approach.

Editing

An important preliminary step in developing data files is the testing of individual records for admissible categories within codes and for internal consistency between codes. When either type of error is encountered, corrective action may be taken by reference to the source document and subsequent updating of the stored record.

Consider the three variables age, sex, and cause of death in a mortality coding system. With the four-digit International Statistical Classification for the latter, only 16 percent of the digits between 1 and 9,999 are assigned to the physician's statement of underlying cause, the remaining 84 percent constituting gaps or inadmissible code categories. Certain causes are sex limited or age limited, or both. Suppose the three variables are recoded as follows:

	Decimal	*Binary*	
Cause (C)	0	0	admissible code no sex or age test
	1	1	impossible code
	2	10	male limited
	4	100	female limited
	8	1000	age under 1 year
	16	10000	age over 1 year
Sex	1	1	impossible code
	2	10	female
	4	100	male

Age--------	1	1	impossible code
	8	1000	age over 1 year
	16	10000	age under 1 year

A test of a single proposition will suffice to identify either impossible codes or impossible combination of codes—

$$P = BOOL[C+SEX + AGE + (C*SEX) + (C*AGE)]$$

the symbols (*) and (+) representing bit union and intersection. If P is not equal to zero, at least one error will have been encountered and the record will be written out for subsequent correction. Extension of the method to additional variables and combinations of variables is direct. The essence of the approach is the setting up of a procedure to recode each variable being tested either singly or in combination. If one is working with a machine that has a large enough memory to store the necessary tables the lookup method is a direct and therefore efficient method for effecting the recode. With smaller machines, variables with a great number of possibilities and gaps in the coding, such as cause of death, could be recoded with sequential comparisons with each entry in a table containing only the admissible codes and an assignment of the recode for the residual category if the table is exhausted without the occurrence of an equal condition.

Applications

During the past year, a series of algorithmic language programs have been written, tested, and used to produce tabular output with the binary coded birth and death files as input (tables 2 and 3). While the major goal has been methodological, the tabulations themselves have intrinsic interest and serve as the basis of reports and publications. Preliminary to a study of the relation between hardness of water supplies and cardiovascular mortality, a table of mortality rates for 16 causes, 10 age groups, sex, and 60 geographic subdivisions of the State was produced, using an input sample of 250,000 death records. A total of 19,200 frequencies, 19,200 age-sex-cause-place specific rates, and 1,920 age-adjusted rates were produced in 30 minutes, using a nonbuffered, 32-K memory binary computer with a character transmission speed of 15-KC and cycle time of 12 microseconds. The same result was achieved in less than 5 minutes on a more advanced 32-K computer with buffering, a 90-KC character transmission speed, and a 2-microsecond cycle time. With either computer, the total production cost, including printing of tables off-line, was less than $100.

In a study of the relationship between perinatal and infant mortality and selected parental and gestational characteristics, a 50,000-cell frequency matrix and a 50,000-cell mortality rate matrix was produced from the 1960 merged birth file in less than 20 minutes, using a binary computer with a 12-microsecond cycle time and 15-KC transmission speed at a cost of $60. The same end could have been accomplished on more advanced computers with up to a 10-to-1 time reduction for a net cost of about $40 at current commercial rates. Similar results have been achieved with studies of deaths by cause of death, sex, month of birth, and birth cohort; of births relating the incidence of congenital malformations to parental age variations; of the concordance of metric and of qualitative traits within twin sets; and of births whereby logistic functions were fitted to birthweight distributions within population subsets.

Discussion

A comparison of the production of tabular output from vital record files by mechanical tabulators and clerical means, by data-processing computers, and by large-memory binary computers, suggests that the latter is the most efficient approach on a cost and real time basis. Standard monthly, quarterly, and annual vital statistics reports for a single State of average size, on the type of binary equipment under discussion, can be produced in a total of a few hours per year. The economics of the situation are such that the vital records tabulation application constitutes a minor part of the justification for an in-house computer, particularly if large-scale binary equipment can be obtained elsewhere on an hourly basis. Proximity, access, and system charges enter the picture. One or more such installations are located in major cities throughout the country.

Of equal, if not greater, importance than equipment considerations are those relating to programing and realization of potential applications. The major limiting factor in the so-called man-machine system is the man. General statistical program packages, such as those written at the University of California, Los Angeles, and at the National Bureau of Standards, represent one useful step in the direction of making computer technology widely available. Of more significance is the development of compiler languages which, in terms of their vocabulary, syntax, and grammar, are readily accessible to the nonspecialist.

It has been an instructive experience to attempt to process massive data files and to develop frequency tabulations therefrom, working on a part-time basis and with only a rudimentary knowledge of computer operations. While some may charitably refer to the effort as "muddling through," the surprising fact remains that the methods described in the preceding paragraphs are easily manipulated by nontechnical persons and result in object programs within a tolerable range of cost and speed. We are approaching that point where the need for data processing can be satisfied by the user himself.

Summary and Conclusions

In the office of biostatistics of the New York State Health Department, the essence of the approach to the use of computer methods to produce tabulations from vital records and other sources of mass data is encoding of a vector label and application of a mixed-base arithmetic. Binary coding and Boolean algebra are used in file development, file maintenance, and data abstraction.

All problem formulation and programing is performed on a part-time basis by statisticians who have no formal training in computing techniques. Algorithmic languages are used exclusively.

The object programs produced are of suitable efficiency as to cost and time.

A methodology has been developed which enables more thorough use of mass data sources.

American Board of Preventive Medicine Examinations

November 1, 1964, is the final date for filing applications for the 1965 examinations for certification as Diplomates in Public Health, Occupational Medicine, General Preventive Medicine, and Aviation Medicine.

The dates, or probable dates, and locations of the examinations follow.

Public Health: Spring 1965 (probably latter half of March), at school of public health of applicant's choice.

Aviation Medicine: April 23–25, 1965, Hilton Hotel, New York City.

Occupational Medicine: April 3–5, 1965, Americana Hotel, Bal Harbour, Miami Beach, Fla.

General Preventive Medicine: part I, written examination, Spring 1965 (probably latter part of March), at school of public health of applicant's choice (part II, oral examination, October 3, 1964, New York City; successful completion of part I required: persons eligible will be notified of exact time and place).

Address inquiries on eligibility requirements for examinations and on fees, examination dates, and requirements for certifications to American Board of Preventive Medicine, Inc., John C. Hume, M.D., Secretary-Treasurer, 615 North Wolfe Street, Baltimore, Md., 21205.