DEMOGRAPHY Volume 9, Number 3 August 1972

PREPROCESSING THE 1960–1970 U. S. CENSUS PUBLIC USE SAMPLES

James M. Sakoda
William J. Sakoda
Department of Sociology, Brown University, Providence, Rhode Island 02912

Abstract—This is a report on computer programming undertaken to preprocess the 1960–1970 U. S. census public use samples to increase their accessibility. This includes wholesale recoding into positive integers from 1 to N, with 0 reserved for "not applicable", combining household and person records, sorting records into ten 1/10,000 samples, compacting binary codes to fit on a single reel of tape, and the production of a revised set of formatted tapes.

The U. S. Bureau of the Census is releasing 1/100, 1/1000 and 1/10,000 samples of household and individual records from the 1970 census and also producing a 1960 version to match as closely as possible the 1970 one (U. S. Bureau of the Census, 1971a,b). There has been some discussion of finding funds to process individual records from previous censuses, and in all likelihood such public use samples will be available from all future censuses. Sophistication in the coding used by the Census Bureau has increased between the issuing of the first version of the 1/1000 sample for the first time following the 1960 census (1963) and the present 1960 public use sample designed to match the forthcoming 1970 samples. Nonetheless, there are some bothersome features which can stand recoding. There is also need for investigation of other considerations, such as blocking, sorting, and compacting of the data, to lower the cost of storage, transmission, and retrieval of the increased amount of data. Based partially on our previous experience in recoding the 1/1000 1960 sample (Sakoda, 1964), we are again undertaking a recoding of at least the 1/1000 public use samples and exploring further means of increasing the efficiency of large-scale data handling. This is a report on the recoding and compression operations, which will make it possible to store the three reels of the 1/1000 samples on a single magnetic tape reel.

The term "public use samples" was introduced with the 1970 census publications to distinguish them from tabulated census reports referred to as "summary tapes". The essential difference between the two is that public use samples provide information on individuals and households, while summary tapes provide tables of age–sex distribution, income by home ownership, etc. for geographical areas such as states, counties, cities, tracts, block groups, and blocks. Public use samples can be used to create any desired table (which may not be published in tabulated form), but only gross identification of area is possible. To study a particular locality summary tapes are useful; to study relationships among selected variables public use samples may be needed. Interested readers are referred to the Census Users' Guide for more detailed information.

Coding

By coding we refer to the assignment of a numeric or alphabetic symbol to information from a survey or census. Such codes are generally punched on cards, and then transferred from cards to magnetic tapes.

There are a few general practices in coding which, when followed, ease the task of data processing. First, we assume that the primary means of analyzing census type of data is by means of cross-tabulation of frequencies in contingency tables. This assumption does not preclude the use of coded data for other types of analysis of continuous data, such as analysis of variance or correlations. The most convenient codes for cross-tabulation are all-integer codes which start at 1, run consecutively, and end with the maximum code, N. A set of such codes can refer to rows or columns in a contingency table and the desired cell for tabulation of a case can be easily located by means of subscripts or through calculation of a subscript. The Ith row and Jth column of a table, IX, for example, can be located as $IX(I, J)$ or the subscript calculated as $(J - 1)*NI + I$, where NI is the number of rows. Any other set of codes is likely to require recoding prior to cross-tabulation, which is an additional expense that should be avoided if possible at the time data are coded (Gittelsohn and Senning, 1964). Use of decimal points and alphabetic codes should be avoided, if possible.

Another problem in coding is the handling of missing data and "not applicable" cases. There are at least three common ways of handling this problem. The first is to leave a blank or a zero when data are missing or not applicable. When an integer variable is read with a FORTRAN format statement a blank and a zero are both treated as zero, and it is not a simple matter to distinguish between the two. Hence, when a blank is used as missing data or not applicable code, a meaningful zero should not also be used for the same variable. The desire to use a zero undoubtedly arises from the desire to maximize the number of codes per card column. By using a zero as a meaningful code it is possible to increase the number of integer codes per card column from nine to ten. If this is done, however, there remains the problem of distinguishing a blank from a zero. The trouble in making this distinction is not trivial—it is necessary to read in the same variable both in integer and alphameric formats using a reread facility (e.g., the T format). Whenever a zero integer is encountered, it is necessary to check the alphameric code to determine whether it is a zero or a blank.

Another solution to the "not applicable" category is to assign the next higher code above the maximum meaningful code. Thus, for "worked last year" there would be three possible codes, 1 for "yes", 2 for "no", and 3 for "not applicable". One of the nuisances of such coding, although not a major one, is that "not applicable" is represented by a code changing from one variable to another. To avoid this, some researchers assign a 9, 99, or 999 as the missing or nonapplicable code. This has the advantage of having an easily identifiable code for coding purposes. In machine processing the data, however, it becomes a source of programming difficulty. A frequency distribution for "worked last year", for example, instead of three categories, including one for "not applicable", might appear with nine categories, with the third through eighth categories empty. Here again the problem can be overcome through programming, but the device becomes a stumbling block. Furthermore, individual cases, such as infants, with much missing information may end up with many 9 punches along the bottom row of the card, which weaken it. Another reason for avoiding the use of 9's for missing codes is that it hinders any compression operation dependent upon keeping codes

numerically as small as possible. A compromise approach is to use 9's for coding purposes, but to change them at the time codes are put on tape.

Clearly, the use of blank or zero for missing data or the not applicable category is the preferred practice, particularly after one has moved from card codes to magnetic tapes, where a record can consist of as many columns as one needs and one is not restricted by the finite length of a card. Blanks or zeros can then be treated uniformly as data to be ignored during tabulation. There are occasions, however, when a meaningful zero is better retained, as in recording a count, such as the number of children.

For the majority of its variables the Census Bureau has used codes from 0 to N, where N is the maximum code and is very often the "not applicable" category. Because of the practice of allocating (i.e., filling in) missing information there is no need to distinguish between "missing" and "not applicable". Hence, the Census Bureau could have easily used 0 or blank as its "not applicable" category and started meaningful codes from 1 without increasing the number of codes per column, in most instances. The most common recode operation we used was one which changes codes from 0 to $N - 1$ to new codes from 1 to N and codes N as 0 when it refers to "not applicable". On some occasions the Census Bureau has used more than one "not applicable" category, but we changed these to 0 because the information (such as group quarters or Form Number) was available elsewhere in the record anyway. This change from starting with 0 to starting from 1 caused only one increase in the number of columns. For Columns 72–73 of the household record, Head's Age, the Census Bureau's code of "01" for "not applicable" was recoded to "0" and "00" for 100 or more was changed to "100". This increase was offset by eliminating the unused Column 69.

Non-Numeric Codes

With only a few exceptions the Census Bureau employed all-numeric integer codes for their new public use samples, an improvement over the codes used for the first 1/1000 1960 sample, which included many "+" and "−" punches. These occasional non-numeric punches required that the column be read in both integer and alphameric (A) format. Only two of these now remain. In the second column of the personal record, Detailed Relationship, a code of "−" is used for what might be a nonapplicable category. During our recoding operation this is read in the alphameric mode and all "−" codes are assigned an integer 10. This entailed an increase by one in the number of columns required. Later this 10 is changed to "0" since it is a nonapplicable category, while the 0 is changed to 1. In three places in the personal record and one in the household record in coding income a "−00" is used to represent losses of $1–99. These variables were also read in both integer and alphameric format and all "−00" were assigned a −100, a code that happened to be unused in the coding scheme. During the recode phase new codes were assigned to income in order to get rid of negative codes. The frequent use of non-numeric codes and shared items in the 1960 1/1000 tapes undoubtedly originated from card rather than magnetic tape technology. The use of card sorting equipment, for example, made it possible to use double punches in a column and furthermore required that as much information as possible be punched on a single card.

Shared Items

Item sharing was used in several places in the initial coding, but only one of these remained in the 1960–1970 public use sample version. Person Record Columns 15–19 represented State of Birth if Column 14, Place of Birth, was equal to 1 (U. S. State). Otherwise, Columns 15–16

contained codes for Country of Birth. Since one set of codes ran from 01–56 and the other from 01–99 they could easily be confused in the course of tabulation, unless controlled for Place of Birth. The items were unshared by changing the codes for Country of Birth to 57–154. This caused the third and last increase in number of columns. The second and third increases were adjusted by eliminating Columns 87–88, which were not being used.

Reordering Codes

In a number of cases it was possible to improve the order in which codes were assigned. Two basic reasons for reordering were the maintenance of proper order when a variable could be ordered in a sensible manner and the avoidance of gaps in codes in the middle of the series. In attempting to match the 1960 and 1970 versions of the public use sample the Census Bureau found it convenient to create gaps in the 1960 codes. The coding for the 1960 version was made to correspond as closely as possible to the 1970 one by leaving gaps in codes when necessary. From a processing point, however, it is more convenient to have these gaps come at the end of a line of a table rather than in the middle.

An example of improving the order of codes is Air Conditioning (Housing Column 59). The codes and recodes are shown in Table 1.

Since we have something like an ordered variable it is preferable to code No Air Conditioning before One-Room Unit and Two- or More Room Units. Similar changes were made for others such as Television and Radio. Both not applicable categories were assigned "0", reducing the maximum number of codes from five to four.

TABLE 1.—Codes and Recodes for Air Conditioning

Description	1960 Code	Recode
1 room unit.	0	2
2 or more room units	1	3
Central system . . .	2	4
No air conditioning.	3	1
NA (20% sample). . .	4	0
NA (G.Q.).	5	0

The Census Bureau used codes for income which allowed one coded unit to represent $100 of income, representing losses with negative codes. The use of negative codes introduces complexities in data processing, and we have chosen to treat income as a set of ordered categories from maximum loss to maximum income, ignoring the effort to use $100 as a standard unit. The codes and recodes are shown in Table 2.

The gap in the 1960 coding can be illustrated by Household Record Column 27, Tenure, as shown in Table 3. In the 1960 coding Owned and Being Bought are grouped together so that the code of 1 is not used. This awkward gap in the coding is eliminated by moving Owned or Being Bought to the last line of the table. In the 1970 version Being Bought will be given an additional line, i.e., a code of 4.

Two-Digit Recodes

There was enough space in the 1960 and the 1970 15 percent sample version to provide two-digit recodes for four variables with three-digit codes. These included family income, industry, occupation, and total personal income. This, of course, is of convenience to users.

Combining Household and Person Records

In the first version of the 1/1000 census tapes household and person information were combined into a single record. On this 1970 round household records are separated from person records. Each household record is followed by one or more person records to which the household record applies. There are roughly 58,000 household records in the 1/1000

TABLE 2.—Codes and Recodes for Income

Description	1960 Code	Recode
No income.	999	101
$1-99	000	102
$100-199 to $9900-9999	001 to 099	103-201
$10,000-10,999 to $24,000-24,999 . . .	105 to 245	202-216
$25,000 or more.	250	217
$1-99 loss	-00	100
Loss in hundred dollar intervals . . .	-01 to -98	2-99
Loss of $9900 or more.	-99	1
NA (under 14 years of age)	998	0

sample for 1960 and approximately 180,000 person records. Each record, either household or person, consists of 120 characters or bytes, which can be packed 800 to the inch in standard 9-track density. Combining household and person records means an increase in tape storage requirement of about 50 percent. This would have meant an appreciable increase in the number of tapes required, particularly for the 1/100 samples. The 1/1000 samples were put on three tapes and the 1/100 on 30. An increase of 50 percent would have increased both the number of tapes and the speed of reading them. Nonetheless, most computer programs are designed to handle a file of single records and this is the form which most users will find convenient for cross-tabulation or other uses. We have therefore chosen to combine household and person records, relying on some increase in blocking and particularly on compression of records to reduce the total volume on tape. The separation of household records from person records requires that the count of the number of person records be absolutely accurate. We found one error in household No. 32574, which had two person records following it, but a 4 stored in the number of persons cell.

Sorting into Ten Subsamples

The 1/100 sample is organized by states and it must be assumed that the 1/1000 samples also are organized geographically. Each household record has a subsample number from 00 to 99, and the 1/1000 sample for 1960 was selected by picking out households with unit digit of 2. In order to tabulate a national sample, it is necessary to run through the entire 180,000 records. One can of course use the 1/10,000 sample, but this would only leave a choice between 18,000 and 180,000 records. To allow runs in between these numbers we have made what amounts to ten 1/10,000 samples, using the tens digit of the subsample number for each household. The subsampling was done first, and recoding was performed on each of the ten smaller tapes. One of the important uses of subsamples is in the development of error variances by making the same tabulation from two or more different random samples.

To perform the subsampling operation blocked records were read without formatting, and the few words that had to be converted from EBCDIC to binary were done arithmetically. For example, in EBCDIC coding two decimal digits,

TABLE 3.—Codes and Recodes for Household Record Column 27, Tenure

Description	1960 Code	Recode
Owned or being bought	0	3
Rented for cash rent.	2	2
No cash rent.	3	1
NA (G.Q.)	4	0

N and M, are combined with F to form $FNFM$. The two F's can be eliminated through subtraction of $F0F0$. $F000$ is equal to -4096 and $F0$ to 240. $F0F0$ is therefore equal to -3856. Performing the subtraction:

$$N0M = FNFM - F0F0.$$

N and M are then obtained by using integer arithmetic on hexadecimal numbers:

$$N = N0M/256$$

$$M = N0M - N0M/256 * 256.$$

To run through two and a half reels of source tapes and to take off a ten percent sample required a little over an hour. Practically all of the time was required to read and write tapes. If FORTRAN formatting had been used the process would have required many more hours.

Formatted vs. Unformatted Tapes

The Census Bureau provides tapes in formatted EBCDIC code for 9-track and in BCD form for 7-track tapes. BCD stands for binary coded decimal and EBCDIC for extended binary coded decimal interchange code. Both are standard ways of representing alphabetic and numeric information in a form which is essentially machine independent. Both represent a decimal digit or alphameric character originating in a card column as a combination of two hexadecimal (or octal in the case of BCD) numbers, such as $F0$ for 0 and $F1$ for 1. For efficiency in storage EBCDIC codes are usually packed so that two or more characters can be stored in a single physical word. If few calculations need to be performed, data can be kept in formatted form. In the case of summary tapes conversion from formatted to unformatted form is not necessary if tables are to be retrieved and printed out without intermediate calculations. To perform calculations on the information it is generally necessary to transform the EBCDIC codes to binary form. This is generally done with a FORTRAN formatted read statement. This is a costly operation requiring large amounts of central processing unit time. One reason for this is that each record must be stored in the FORTRAN input–output buffer and there subjected to extended checks and dispersal to storage locations specified by the READ statement. This process can be bypassed after an initial unformatting run by keeping the records on tape in unformatted or binary word form. The variables are not only on binary form ready for machine calculation, they are also separated into individual words. For our own use we generally keep files of records in unformatted form to rid ourselves of the cumbersome formatting operation. For one thing, this eliminates the need for variable format (i.e., a different format for use of a program on different sets of data).

The move from formatted to unformatted records can either increase or decrease space requirements. Public use sample records are packed into 120 byte records with little loss of space. The combined household and person record, originally consisting of 240 bytes, expands to 176 words or variables, each requiring 16 bits or 2 bytes of storage on tape, or 352 bytes. This again represents an expansion of about 50 percent. Part of this expansion is accounted for by empty variables (which were only set up for compatibility with the 1970 version) and allocation variables which require only a single character. The gain in processing speed is great enough to make this expansion a negligible factor. It can be compensated for to some extent by increased blocking, and we also count on the compression procedure to more than compensate for this increase.

Blocking

A word needs to be said about blocking. Blocking represents a way of saving space and processing time. When a tape file is created a gap of about .6 inches

is required between reading operations. At a density of 800 bytes per inch a 120 byte record would only require .15 inches of tape, and hence the record gap would be four times as great as the record itself. By writing several records together at the same time it is possible to save both the space and time required by the gaps between individual records. The 1960–1970 public use sample records have been blocked a reasonable amount—a blocking factor of 15, i.e., 15 records of 120 bytes each per block. A block of records then consists of 1800 bytes requiring 2.25 inches at a density of 800 bytes per inch, and about .6 inches is required between blocks. An input or output buffer equal to the size of the block of records is required in order to process blocked records, and hence there is a limit to the amount of blocking one can do to satisfy machines with relatively small amounts of core memory. Even so, it is possible to increase the blocking factor by at least a factor of 2 (i.e., use about 3600 bytes) without pressing excessively on core requirements of most machines. Saving space on tape generally means saving processing time, since it is the time taken to pass the tape through the read head which largely determines the processing time, assuming that additional time is not required for additional chores, such as unpacking.

Special Formatting Routine

The use of highly uniform coded data (e.g., positive integers only and columns which need to be skipped) introduces the possibility of bypassing the FORTRAN formatted read–write routine completely by writing special routines which perform only a limited amount of checking and are designed to operate efficiently on coded data. This would assume, for example, that alphameric information and negative integers were not present. The coding operation on a single 1/10,000 sample required seven hours of IBM 1130 time. Much of this was required by the FORTRAN formatting operation to convert the EBCDIC code to binary form and relatively little for the extensive recoding operations. We have estimated that the creation of a formatted tape from the recoded packed tape has been speeded up by a factor of 12 by using a specially coded formatting routine instead of the FORTRAN formatted output operation. The formatting, which is all in positive integers, is specified by a minus number for skips and a positive number for the number of decimal digits.

Data Compression

When the maximum code for any variable is less than the fixed word size there is a possibility of eliminating space wastage by allotting to each variable only the required maximum number of bits. This amounts to making individual word lengths variable rather than fixed, measuring them in number of bits. An empty word need not take any space. An allocation indicator word can be limited to a single bit. Sex needs only two bits. Heavily coded data requiring only one or two columns of decimal digits can be expected to require on the average between three and four bits. On the 1960 tape, for example, the 176-word combined record of household and person data could be compressed into 30 16-bit words, a ratio of almost 6 to 1. The average number of bits per word was 2.6. This low figure was caused by 42 empty words reserved for use in the 1970 15 percent sample tape and 13 allocation words requiring only a single bit. If these were eliminated, the average number of bits would be 3.65. With similarly coded integer data a compression ratio of at least 4 to 1 could be expected, if 16-bit words were used. With 32-bit words this ratio would be doubled. The recoded tape was blocked 10 records per block. The packed records were blocked 50 records per block, with a single block requiring 3000 bytes or 4.35 inches. The entire 1960 1/1000 file of

180,000 records fits into approximately 1300 feet of magnetic tape, a little more than half a reel. At this rate the entire 1/100 reels of tape could be fit into about seven reels of magnetic tape.

Processing Time

With large files processing time, as well as amount of bulk storage space, becomes a problem. There is often an exchange between space and execution time, but we have estimated that the reading and unpacking of compressed records can be done in about a third of the time it takes to read records which are not compressed. This assumes that unpacking constants are calculated and stored prior to the read and unpack operations, and that only about five machine instructions are needed to unpack each word. The cost here is the storage space for unpacking constants, with six words required for each variable which is unpacked.

Discussion

With the increase in volume of data which needs to be kept in machine readable form there is a corresponding need to increase the efficiency of storage, transmission, and retrieval of data. One of the primary goals is to avoid the use of FORTRAN formatted read operations, which are inherently slow. To this end the use of all integer codes running from 0 to N, where N is kept as small as possible, provides the basis for other simplifications. The storage of data in packed form, eliminating unnecessary empty positions, is desirable provided the unpacking operation is efficient enough. FORTRAN formatted input–output represents such a packing and unpacking operation, but is complicated by having to deal with other forms of data such as alphameric characters and floating point or real numbers. Storing data in decimal form is not necessarily inefficient. The number 12, for example, is stored in EBCDIC as *F1F2* and requires 16 bits. If only integers were involved the *F* could be eliminated and each decimal number stored in four bits or four digits per word. The choice between decimal and binary representation ultimately rests with usage. Decimal representation is convenient for printing, but not for calculations; binary numbers can be used for calculations, but require conversion for printing. Census summary tapes, which are in table form, may be most conveniently handled as decimals; public use samples must undergo some calculations and hence are best stored in binary form. As we have seen, however, storage of the public use sample data in unformatted but unpacked form would expand storage by about 50 percent, but would eliminate the costly use of the FORTRAN formatted input–output operation. Packing coded integers in binary form greatly reduces the amount of storage required, generally by a factor of 4 or 5 to 1 using 16-bit words and double that amount for 32-bit words. Our experience shows that unpacking selected words from a packed tape can be done more efficiently than dealing with the unformatted but unpacked tape. There is an additional advantage in that the entire 1/1000 sample resides on a single reel of tape, while the unformatted version requires at least three tapes.

There is another question that needs to be considered and that is whether storage of data in compressed binary form meets the need of users of different machines. Binary integers are standard for different machines, since they are right-adjusted within the word and consist of zeros and ones. When they are packed they cut across word boundaries, and hence they are to some extent independent of word lengths. It was possible to use a 360 utility program to rewrite 9-track packed words into 7-track packed words. This required use of two 7-track tapes, because the density dropped to 556 bytes per inch and one-third more words were required. Hence,

there are strong arguments for keeping data requiring calculation in compressed binary form. There remains the need to write packing and repacking routines for different machines on which compressed tapes are to be used.

The question of whether it is worthwhile to preprocess the 1960–1970 public use samples can be answered affirmatively. The advantages of the preprocessing can be made available to others by providing copies of the tapes which have been preprocessed and by providing copies of programs used in making the conversion. In addition to the unformatted and packed tapes, there will be a recoded formatted public use sample of the 1/1000 version on three tapes for distribution. The extension of this technology to future censuses and other data banks is obviously desirable.

Acknowledgments

This work was supported by PHS Grant MH-08177, Computer Utilization in the Behavioral Sciences, of the National Institutes of Health, Public Health Services. We thank Professor Albert Chevan of the University of Massachusetts for an opportunity to see his bit-packing program, after which our initial effort was patterned. Those interested in acquiring copies of recoded tapes and programs used in producing them should contact Professor James M. Sakoda.

References

Gittelsohn, A. M., and Senning, B. S. 1964. Tabulation of vital records by computer. Public Health Reports 79:895–904.

Sakoda, James M. 1964. Brown University 1960 Census Code Book. Sociology Computer Laboratory, Brown University, Providence, Rhode Island (mimeo.).

U. S. Bureau of the Census. 1963. U. S. Censuses of Population and Housing: 1960. One-in-a-Thousand Sample Description and Technical Documentation.

———. 1970. 1970 Census Users' Guide, Part I and II (October).

———. 1971a. One in a 100, A Public Use Sample of Basic Records from the 1960 Census (April).

———. 1971b. Public Use Samples of Basic Records from the 1960 and 1970 Censuses. Data Access Description, No. 24 (May).