Behavior Research Methods & Instrumentation
1982, Vol. 14 (1), 37-38

Utility subroutines for data manipulations

HARALD M. RAUSCHER
North Central Forest Experiment Station
Marquette, Michigan 49855

and

GREGORY J. BUHYOFF
School of Forestry and Wildlife Resources
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061

This article describes a series of utility subroutines that, together, define a general data matrix manipulation system for microcomputer applications. The data structure defined by these utilities has been modeled after that used by SAS (Helwig & Council, 1979) and by SPSS (Nie, Hull, Jenkins, Steinbrenner, & Bent, 1975), two popular data analysis systems for large computers. The popularity of SAS and SPSS is due, in part, to their use of a standard data structure to unite a diverse collection of otherwise distinct data analysis programs. Following this pattern, for more than a year, we have been using the data structure defined by the subroutines described in this article as a standard to unify 32 statistical analysis programs into a coherent system (Buhyoff, Rauscher, Hull, & Killeen, 1980).

This set of subroutines is referred to as a "seed" program because it is used as the foundation upon which all other programs are built. A seed program enhances software development productivity because these commonly needed subroutines remain the same from program to program. In addition, the seed program isolates unavoidable system dependencies in a handful of small routines that can be written for a particular operating environment (Kernighan & Plauger, 1976). This practice enhances software portability. The concepts of a standardized data file structure and a set of system-dependent input/output utilities may be profitably tailored to any computer system and language.

The seed subroutines are written in BASIC for the TRS-80 Model I disk-based microcomputer.

Two seed programs are presented, one for sequential disk data files and the other for random files. Sequential access refers to reading data from a disk file or writing data to a disk file "from start to finish," without being able to directly access a particular record in the file (Radio Shack, 1979). The use of sequential access requires that all data be stored in the computer memory at the same time. Consequently, sequential access is limited by the memory size of the computer.

Random access allows the user to directly read or write any record to a file without having to begin reading at the start of the file. Random-access techniques are more difficult to implement than sequential access techniques: Few examples of the use of random-access techniques are available in the microcomputer literature. Random-access files may be used to circumvent the limited rapid-access memory of most microcomputers by shuttling data back and forth between disk storage and memory storage. Although larger sets of data may be manipulated using the random-access techniques, execution speed is substantially reduced because of the frequent need to read data from and write data to a disk.

Standard Data File Structure. The data are built into a matrix, Aij, where i = number of observations and j = number of variables. Each data set has three automatically defined data files. If "TEST" is the name of an example data set, the data files "TESTM," "TESTN," and "TESTL" are automatically created or updated every time "TEST" is saved to disk. File "TESTM" contains the matrix Aij. File "TESTN" contains the ij, row/column indexes, and file "TESTL" contains the names of the variables (i.e., columns of Aij). Data file matrix manipulations are greatly simplified by using this three-file organization.

Sequential Data Seed Program. The subroutines that make up the sequential access seed program perform the following functions: (1) format and control user input, (2) read data from disk, (3) write data to disk, (4) enter data manually from a keyboard, (5) display data on a video screen, and (6) print data on a line printer.

The most basic requirement for any interactive program is a subroutine that accepts input commands from the user, usually via the keyboard. The formatted input control subroutine in the sequential access seed program allows the programmer to restrict input to either numeric or alphanumeric digits and to set the maximum number of digits. The backspace key is used to erase mistaken keyboard entries. The user must press the ⟨ENTER⟩ key to generate a line feed for multidigit input. The line feed is automatic when only a single digit of input is required.

The data matrix "read" subroutine requests the name of the desired data set from the user, loads the row and column dimensions into variables R and C, respectively, loads the variable (column) names into N$(C), and loads the data matrix into variable A(R,C). Every data set read from disk is loaded into Matrix A(R,C), and every data set written to disk comes from Matrix A(R,C). This convention makes the input/output operations of any program based on these seed routines easy to understand.

When the sequential data disk "write" subroutine is used, the user is prompted to enter the data set name and the disk drive number on which the data are to be saved. The data set must be in memory as Matrix A(R,C), with R and C containing the dimensions of Matrix A and N$(C) containing the names of the variables (columns) of Matrix A. If no names are found in N$(C), the variables are named consecutively "1," "2," to "C."

0005-7878/82/010037-02$00.45/0

Data sets may also be entered manually via the keyboard. The dimensions of the data matrix are requested and entered into R and C, followed by the name for each variable. The value of each element is then requested from the user in row-by-column order. The data are loaded into Matrix A(R,C) and may be saved to disk using the disk "write" subroutine.

The video display of Matrix A(R,C) is a window that shows a block of data (13 rows by 4 columns) at one time. This routine is tailored to the 16 row by 64 column display capacity of the Radio Shack TRS-80 Model I computer video display terminal. By pressing the up, down, left, or right arrows on the keyboard, the user has complete control over which portion of the data matrix is on display. Pressing the period (".") key activates a subroutine that allows changing the names of the variables of the data set. Pressing the "CLEAR" key returns the user to the calling program.

The line-printer subroutine is similar to the video display routine. A check is made to ensure that a data matrix is defined in A(R,C) and that the printer is connected to the system and is operational. All rows of the first four columns of the matrix are printed first, followed by all rows of the second four columns, and so forth.

Random-Access Seed Program. The random-access seed program is composed of subroutines that perform the following functions: (1) format and control user input, (2) read/write a data record to disk, (3) enter data manually from a keyboard, (4) display data on a video screen, (5) print data on a line printer, and (6) make a copy of a data set.

The formatted input control module in the random-access seed program is exactly the same as the one in the sequential seed program described above.

The random-access read/write subroutine performs the same functions as the read and write subroutines in the sequential access seed program. A file buffer, located in the direct-access memory of the computer, is used to temporarily store the numbers of interest for any given operation. The most popular disk operating systems for the TRS-80 microcomputer define random-access operations so that data are passed to and from the disk in 256-byte units called records (Radio Shack, 1979). Each single-precision number utilizes 4 bytes, so a maximum of 64 such numbers can be passed by each random-access operation for temporary storage in the Matrix Z1(IS,I2). This matrix, analogous in function to the Matrix A(R,C) in the sequential seed program, is the temporary holding matrix for that portion of the total data set currently being processed by a control program. The variable I2 identifies the column of Z1 that holds the current record of 64 single-precision numbers of interest. IS is used to point to the particular number, of 64, that is called for by the control program.

Data may be entered manually from the keyboard, as in the sequential case. The user is asked to specify the file name of the data matrix and its dimensions. The subroutine then requests the user to input the data, fills the buffer, and saves the filled buffer onto the disk specified.

The video display routine is similar to its sequential counterpart in most respects. Because random data sets may be very large, the user must specify the matrix row/column coordinates of the data window he wishes to see. The arrows on the keyboard function in the same way as in the sequential version to move the view window one page at a time in any direction. Pressing the letter "C" allows the user to skip to another row/column window. Pressing the "CLEAR" key returns the user to the central menu.

Printing the data matrix follows the same pattern as described for the sequential case. Finally, it is frequently necessary to preserve one version of a data set while changing another version. The copy subroutine allows duplication of a data set under another name.

Availability. A detailed programmer's guide that includes source code listings for all subroutines and a discussion of the logic and variables used in the seed subroutines is available upon request. Requests for the programmer's guide and a machine-readable copy should be addressed to Harald M. Rauscher, Northern Hardwoods Laboratory, 1030 Wright St., Marquette, Michigan 49855. Machine-readable copies will only be made if the request is accompanied by a 5.25-in. soft-sectored disk.

REFERENCES

Buhyoff, G. J., Rauscher, H. M., Hull, R. B., IV, & Killeen, K. Microcomputer resident comprehensive statistical analysis. *Behavior Research Methods & Instrumentation*, 1980, **12**, 551-553.

Helwig, J. T., & Council, K. A. (Eds.) *SAS user's guide, 1979 edition*. Raleigh, North Carolina. SAS Institute, 1979.

Kernighan, R. W., & Plauger, P. J. *Software tools*. Reading, Mass: Addison-Wesley, 1976.

Nie, N. H., Hull, C. H., Jenkins, J. G., Steinbrenner, K., & Bent, B. H. *SPSS statistical package for the social sciences* (2nd ed.). New York: McGraw-Hill, 1975.

Radio Shack. *TRSDOS and disk basic reference manual*. Fort Worth, Tex: Tandy Corporation, 1979.

(Accepted for publication November 18, 1981.)